0410

OIPE JC144
FEB 13 2002
PATENT & TRADEMARK OFFICE

COPY OF PAPERS
ORIGINALLY FILED

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

Applicant	:	**Arthur Sheiman, et al.**
Title of Invention	:	**TIME VARYING FILTER WITH ZERO AND/OR POLE MIGRATION**
Filed	:	**September 28, 2001**
Serial No.	:	**09/966,802**
Group Art Unit	:	**Not Yet Assigned**
Examiner	:	**Not Yet Assigned**
Attorney Docket No.	:	**024/46**

09966802 .021302

Box Non-Fee Amendment
Commissioner for Patents
Washington, DC 20231

PRELIMINARY AMENDMENT

Sir:

Prior to examination on the merits, please amend the above-referenced application as follows:

IN THE SPECIFICATION:

Please amend the paragraphs appearing from Page 11, line 6 through and including Page 13, line 9, to read as follows.

-- An example system implementing the notch filter depicted in Fig. 1 can easily be conceived, where the filter is bypassed, but never removed (thus neither removal of the filter from the final configuration shown in Fig. 3, nor the steps, to being fully implemented, as depicted in Figure 1. Once the final step has been reached, i.e., STEPS=300 [in the illustrative system of Fig. 6], the [state] system moves [, along path 611, to state 620] to the third state, "Notch ON." From there a Notch[ ]Request = OFF will move the system [, along path 621, to state 630,] to the fourth state "Migrate Notch OUT", where the reverse happens, and the poles and zeros in the configuration of Fig. 1 now migrate to the configuration of Figure 2, via the 300 intermediate steps. When, in [state 630,] the fourth state STEPS =300, the filter is totally migrated out, and the system moves [, along path 631] back to the first state, [to state 600] Notch BYPASS.

As well, if in the middle of a filter migration one way or the other the value of Notch[ ]Request [value] changes, the system will move from [state 610 to state 630] the second state "Migrate Notch IN" to the fourth state "Migrate Notch OUT", or vice versa. [With reference to Figure 6, if in state 630, where] If when the filter is migrating OUT due to a Notch Request = OFF signal, the variable Notch[ ]Request switches to ON, the system moves[, along path 632, to state 610,] from the fourth state to the second, and migrates the filter back IN, beginning from whatever the value of STEP was when the Notch[ ]Request signal changed from OFF to ON. Conversely, if in the middle of a migration IN, where the system is in [state 610] in the second state "Migrate Notch IN", the variable Notch Request goes to OFF, the system moves[, along path 612 to state 630] to the fourth state "Migrate Notch OUT", and the migration OUT proceeds from whatever the value of STEP was in the second state [610].

removal of the filter from the configuration of Figure 2, occurs in this example system). Such a system would have four possible states: one where the filter is bypassed, a second where it is being implemented, a third where it is fully implemented, and a fourth where it is being removed. Once fully removed, the system returns to the first state where the filter is bypassed. A binary variable, for example "NotchRequest" would be an input to such a system, determining whether the filter is to be bypassed or implemented. As described above, each particular application, as well as the subjective perception of the users interacting with the output signal, will determine the minimum steps needed to fall below the threshold of perceptible artifacts (auditory or otherwise, as determined by the signal being processed). Such step numbers can be empirically found with relative simplicity. The number of steps used would be a variable, say "STEPS", where STEPS is the input to a coefficient calculation function, for example, "CalcCoeff." CalcCoeff is determined by parameterizing the migration paths, and could be set such that, for example, STEPS = 300.

Consider the migration of the zeros of the depicted fourth order notch filter of Figure 1. In this case, the poles and zeros align on straight lines in the z-plane, which are the dotted lines 100 in Figure 1. The co-efficients can be calculated in real time, or if the filter used in the system is known ***a priori***, as well as the desired step number ("Steps" in the example system), the various coefficients for each step can be stored in a lookup table, moving thereby the complex high MIPS calculations to non-real-time at the expense of a small to moderate table.

Consider the migration of the zeros of the depicted fourth order notch filter of Figure 1. In this case, for a notch filter, the poles and zeros align on straight lines in the z-plane, which are the dotted lines 100 in Figure 1. The co-efficients can be calculated in real time, or if the filter used in the system is known ***a priori***, as well as the desired step number, the various coefficients for each step can be stored in a lookup table, moving thereby the complex high MIPS calculations to non-real-time at the expense of a small to moderate table.

In [Figure 6 the state 600] the abovedescribed exemplary system, a first state "Notch [Bypass] BYPASS" (all variable and state names being provided merely as examples), is [that] where no filter is operating, and the poles and zeros are colocational, as in Fig. 2. In [state 620] a second state "Migrate Notch IN" the system migrates the notch filter in by moving the zeros of the configuration of Fig. 2 to the configuration of Fig. 1. In a third state "Notch ON" the filter is fully engaged. [States 610 and 630 represent the migration stages "Migrate Notch IN" and "Migrate Notch OUT."] [In state 610] In a final state "Migrate Notch OUT" the filter is migrated from a neutralized state to the fully engaged state, via the intermediate steps. In state 630 filter is migrated from a fully engaged state (as depicted in Fig. 1), to the neutralized state (as depicted in Fig. 2), via the intermediate steps. The number of intermediate steps is controlled by the variable "STEPS."

The system moves from [state 600 to state 610, along path 601] first state Notch BYPASS to second state Migrate Notch IN, when Notch[ ]Request is set to ON. [In state 610] Again, in the second state the filter is migrated from being bypassed (i.e., the poles and zeros are colocational), as depicted in Figure 2, through the 300 intermediate

In the abovedescribed exemplary system, a first state "Notch BYPASS" (all variable and state names being provided merely as examples), is where no filter is operating, and the poles and zeros are colocational, as in Fig. 2. In a second state "Migrate Notch IN" the system migrates the notch filter in, by moving the zeros for the configuration of Fig. 2 to the configuration of Fig. 1. In a third state "Notch ON" the filter is fully engaged. In a final state "Migrate Notch OUT" the filter is migrated from a neutralized state to the fully engaged state, via the intermediate steps. In this final state the filter is migrated from a fully engaged state (as depicted in Fig. 1), to the neutralized state (as depicted in Fig. 2), via the intermediate steps. The number of intermediate steps is controlled by the variable "STEPS."

The system moves from first state Notch BYPASS to second state Migrate Notch IN, when NotchRequest is set to ON. Again, in the second state the filter is migrated from being bypassed (i.e., the poles and zeros are colocational), as depicted in Figure 2, through the 300 intermediate steps, to being fully implemented, as depicted in Figure 1. Once the final step has been reached, i.e., STEPS=300, the system moves to the third state "Notch ON." From there a Notch Request = OFF will move the system, to the fourth state "Migrate Notch OUT", where the reverse happens, and the poles and zeros in the configuration of Fig. 1 now migrate to the configuration of Figure 2, via the 300 intermediate steps. When, in the fourth state STEPS=300, the filter is totally migrated OUT, and the system moves back to the first state, "Notch BYPASS."

As well, if in the middle of a filter migration one way or the other the value of Notch Request changes, the system will move from the second state "Migrate Notch

IN" to the fourth state "Migrate Notch OUT", or vice versa. Where the filter is migrating OUT due to a NotchRequest = OFF signal, and the variable NotchRequest switches to ON, the system moves from the fourth state to the second, and migrates the filter back IN, beginning from whatever the value of STEPS was when the NotchRequest signal changed from OFF to ON. Conversely, if in the middle of a migration IN, where the system is in the second state "Migrate Notch IN", the variable NotchRequest goes to OFF, the system moves to the fourth state "Migrate Notch OUT", and the migration OUT proceeds from whatever the value of STEPS was in the second state. --

REMARKS

This Preliminary Amendment is made to amend certain paragraphs of the specification to cancel references to an unused figure (Fig. 6). The text has been amended to contain the description originally included, albeit references to Figure 6 have been eliminated. A marked up copy of the amended paragraphs is provided as an Exhibit hereto.

Applicant's undersigned attorney may be reached in our Woodbridge, New Jersey office by telephone at (732) 634-7634. All correspondence should continue to be directed to our below listed address.

Respectfully submitted,

KAPLAN & GILMAN, LLP
900 Route 9 North
Woodbridge, New Jersey 07095
Telephone (732) 634-7634

DATED: January 10, 2002

[illegible]
Aaron S. Haleva
Reg. No. 44,733

\Dialogic-Intel\Amendments\Preliminary Amendment - 09-966,802.doc

CERTIFICATE OF MAILING

I hereby certify that this correspondence is being deposited with the United States Postal service as first class mail, in a postage prepaid envelope, addressed to Box Non-Fee Amendment, Commissioner for Patents, Washington, D.C. 20231 on January 10, 2002.

Dated January 10, 2002

Signed [illegible]

Print Name Paula M. Halsey

09966802.011302

OIPE JC144
FEB 13 2002
PATENT & TRADEMARK OFFICE

COPY OF PAPERS
ORIGINALLY FILED

PRELIMINARY AMENDMENT – 09/966,802

MARKED UP VERSION OF AMENDMENTS TO SPECIFICATION TO SHOW CHANGES

[Figure 6 depicts a state machine diagram for the example] An example system implementing the notch filter depicted in Fig. 1 can easily be conceived, where the filter is bypassed, but never removed (thus neither removal of the filter from the final configuration shown in Fig. 3, nor the removal of the filter from the configuration of Figure 2, occurs in this example system). Such a system would have four possible states: one where the filter is bypassed, a second where it is being implemented, a third where it is fully implemented, and a fourth where it is being removed. Once fully removed, the system returns to the first state where the filter is bypassed. A [The] binary variable, for example "Notch[ ]Request" [is the] would be an input to [the state machine] such a system, determining whether the filter is to be bypassed or implemented. As described above, each particular application, as well as the subjective perception of the users interacting with the output signal, will determine the minimum steps needed to fall below the threshold of perceptible artifacts (auditory or otherwise, as determined by the signal being processed). Such step numbers can be empirically found with relative simplicity. The number of steps used [is the] would be a variable, say "STEPS", where STEPS is the input to a coefficient calculation function, for example [labeled] Calc[ ]Coeff [in Fig. 6]. [In the system of Fig. 6, the] Calc[]Coeff [function] is determined by parameterizing the migration paths, and could be set such that, for example, STEPS=300.

09966802.021302